

## Liste ordonnée

Logique permettant de déplacer un objet d'une liste à n'importe quelle position, tout en gardant la cohérence des positions (pas de place vide dans la liste).

Pour cet exemple, on utilise une liste d'utilisateurs stockés dans une base de données MySQL, mais la logique reste la même quel que soit le container.

Soit la table User:

User		
id ↓	name	pos
1	Alain	4
2	Barak	6
3	Boris	5
4	Elise	2
5	Gustave	3
6	Maia	1
7	zachary	7

User		
id	name	pos ↓
6	Maia	1
4	Elise	2
5	Gustave	3
1	Alain	4
3	Boris	5
2	Barak	6
7	zachary	7

### Déplacement d'un user :

Alain ( \$id = 1 ) est actuellement en position 4 ( \$pos = 4 )

Il y a 2 cas possibles : déplacement vers le haut ou vers le bas de la liste.

#### **1. Déplacement vers le haut :**

##### **déplacer cet user en première position ( \$pos\_new = 1 ) :**

- a) Décaler tous les user de la position 1 ( \$pos\_new ) à 3 ( \$pos - 1 )  
pour libérer la position 1 :

**req1** : UPDATE User SET pos = pos+1 WHERE pos >= \$pos\_new AND pos < \$pos

- b) Assigner la nouvelle position au user Alain :

**req2** : UPDATE User SET pos = \$pos\_new WHERE id = \$id

#### **2. Déplacement vers le bas :**

##### **déplacer cet user en position 6 ( \$pos\_new = 6 ) :**

- a) Décaler tous les user de la position 6 ( \$pos\_new ) à 5 ( \$pos + 1 )  
pour libérer la position 6:

**req1** : UPDATE User SET pos = pos-1 WHERE pos <= \$pos\_new AND pos > \$pos

- b) Assigner la nouvelle position au user Alain :

**req2** : UPDATE User SET pos = \$pos\_new WHERE id = \$id

1.

User		
id	name	pos
6	Maia	2
4	Elise	3
5	Gustave	4
1	Alain	4
3	Boris	5
2	Barak	6
7	zachary	7

req2

2.

User		
id	name	pos
6	Maia	1
4	Elise	2
5	Gustave	3
1	Alain	4
3	Boris	4
2	Barak	5
7	zachary	7

req2